JUIN 1990 - No 1 nouvelle édition (no 54 ancienne édition)



JOURNAL D'ETUDES ET DE DOCUMENTATION EN INFORMATIQUE INDUSTRIELLE

EDITORIAL

Nombreux sont les ingénieurs regrettant qu'il n'y ait pas, en informatique industrielle, l'équivalent de revues telles que Microsystèmes, Minis et Micros, Doctor Dobb's, etc...

C'est pour combler cette lacune qu'a été créé ce Journal d'Etudes et de Documentation en Informatique Industrielle. Présenté sous forme de lettre-magazine, il se veut avant tout pratique et pragmatique. Aussi ne vous attendez pas à trouver ici une énumération à la Prévert des x derniers servo-mécanismes ou logiciels présentés sur le marché. Des études ciblées, une explication claire des concepts mis en oeuvre dans certains produits, des fiches techniques permettant de se repérer facilement au milieu du foisonnement de matériels et logiciels proposés par le marché, voilà quelques unes des clés qui ont guidé notre démarche.

Axé temps réel et contrôle de processus, JEDII se veut votre journal, c'est-à-dire un outil réellement utilisable par chacun et surtout lui permettant "de ne pas réinventer la roue" !

Nous comptons sur vos idées, vos critiques, en bref votre collaboration, pour faire de cette lettre un trait d'union entre tous ceux qui conçoivent et utilisent les produits d'informatique industrielle. Sa taille n'est donc pas figée mais évoluera en fonction de vos demandes, de vos conseils, et (pourquoi pas ?) de vos articles.

Aussi n'hésitez pas à nous écrire, à nous téléphoner, ce pour nous demander de parler d'un environnement ou d'un dispositif particulier ou encore pour nous expliquer les problèmes que vous avez rencontrés dans la mise en oeuvre de telle ou telle "solution".

La réponse existe forcément quelque part. A nous de la trouver, à nous d'être à votre écoute.

SOMMAIRE

ENQUETE LECTEUR

Afin de pouvoir mieux répondre à vos besoins, JEDII vous prie de bien vouloir répondre à ce questionnaire qui nous permettra de mieux vous connaître et d'appréhender plus finement vos sujets de préoccupation. Merci de bien vouloir le retourner ou le faxer à :

JEDII - REM CORP Publications 17, rue de la Lancette 75012 Paris Fax : 43 42 32 15

Nom:.. Prénom:..........................
Fonction:...
Société:..
Domaine(s) d'activité:..
Adresse:..
Code postal:............ Commune:..
Tel:........................... Fax:.........................
Taille de l'entreprise: 1 à 20 20 à 50 50 à 100 plus de 100 (rayer les mentions inutiles)

Utilisez-vous: des automates programmables: oui non Si oui, lesquels:............................
des réseaux locaux industriels : oui non Si oui, lesquels:...........................

Quels sont les sujets vous intéressant plus particulièrement? (donner un numéro par ordre de préférence, de 1 à 10, exemple: 10 très grand intérêt, 1 intérêt superficiel)

.... Contrôle temps réel
.... Robots
.... Automates programmables
.... Micro-contrôleurs
.... Vision assistée par ordinateur
.... Informatique embarquée
.... Systèmes experts industriels
.... Ateliers flexibles
.... CFAO
.... IAO
.... Modélisation par ordinateur
.... Ordonnancement
.... Gestion de production
.... Capteurs
.... Réseaux locaux industriels
.... Mathématiques appliquées
.... Reconnaissance optique
.... Autres, précisez:...

Décrivez en quelques lignes les sujets que vous désireriez voir abordés:................................
....
....

Concevez-vous des produits (matériel ou logiciel) susceptibles d'intéresser JEDII? oui non

JEDII, Journal d'Etudes et de Documentation en Informatique Industrielle est une lettre d'informations mensuelle publiée par REM CORP SARL au capital de 150.000 Frs. RCS : B 352 959 092 SIRET : 959 092 00011
Siège social, rédaction: 17 rue de la Lancette 75012 PARIS. Tel : (33) 1 43 40 96 53. Fax : (33) 1 43 42 32 15
Directeur de la publication : Michel Rousseau. Rédacteurs en chef : Marc Petremann, Michel Rousseau.
Commisssion paritaire : en cours. Dépôt légal imprimeur : 2° trimestre 1990. Imprimé en France.
Il a été tiré 6.000 exemplaires du présent numéro.

ABONNEMENT

Nom:.. Prénom:..........................
Fonction:...
Société:..
Domaine(s) d'activité:..
Adresse:..
Code postal:............ Commune:..
Tel:........................... Fax:.........................

désire souscrire un abonnement pour 10 numéros dont un numéro double: 1400 FF TTC. Offre spéciale de souscription (valable jusqu'au 15 septembre 1990) **850 FF** TTC.

Date:....... .... Signature:

Paiement à l'ordre de REM CORP. A adresser à REM CORP 17 rue de la Lancette 75012 PARIS

LE CONTROLE TEMPS REEL

LA FAMILLE RTX2000 DE HARRIS
UN NOUVEAU CONCEPT POUR LE TEMPS REEL

QUELLES SONT LES EXIGENCES DU TEMPS-REEL ?

"Qu'y a-t-il d'assez rapide pour le temps-réel?". La complexité de cette question prend toute sa dimension lorsque l'on aborde la nature spécifique d'une application temps-réel. Une réponse acceptable dans le cadre d'un système transactionnel peut être de l'ordre d'une demi-seconde. Pour un système de contrôle avionique, il est probable qu'un temps de réponse du même ordre soit trop lent.

Dans le premier cas, les processeurs conventionnels de traitement de l'information apportent une solution efficace aux besoins temps-réel. Dans le second cas, et dans d'autres, comme l'acquisition de données, le contrôle de processus, la robotique, les réseaux locaux et le traitement numérique, répondre en un temps donné est extrêmement important.

Il apparaît donc que le temps-réel ne peut être défini que dans le contexte de l'application finale. Les besoins de traitement doivent donc être choisis afin de répondre aux événements extérieurs critiques, ce en terme de temps. Nombre d'ordinateurs nécessitent une grande vitesse d'exécution des instructions. Pour les applications temps-réel, la latence aux interruptions et le changement de contexte sont des spécifications importantes et s'ajoutent au besoin de vitesse d'exécution.

Les progrès récents, tant au niveau de l'électronique que des logiciels, ont permis de réduire le temps de réponse aux interruptions et le changement de contexte à quelques microsecondes sur les processeurs actuels.

Le microcontrôleur 16 bits RTX2000 le réduit à 400 nanosecondes pour les interruptions, à deux microsecondes pour un changement de contexte, et atteint une performance de 10 Millions d'instructions par seconde.

La considération clé dans le traitement temps-réel reste la prédictibilité. La plupart des ordinateurs possèdent des mécanismes propres à diminuer le temps d'exécution moyen des instructions. Ces mécanismes, *pipelines*, *mémoire cache*, *registres intégrés* et *compilateurs-optimiseurs* contribuent tous à l'amélioration du temps moyen d'exécution. Ils contribuent également à accroître l'incertitude en matière de prédictibilité du temps d'exécution.

Finalement, on a recours à la logique externe, à savoir des contrôleurs de DMA ainsi que des processeurs d'E/S spécialisés pour faire face aux besoins les plus simples d'interfaçage avec le monde extérieur.

Grâce à sa vitesse d'exécution, à ses réactions rapides aux événements extérieurs et à la prédictibilité de ses temps de traitement, le RTX2000 contribue de manière significative à une meilleure intégration de l'application, en remplaçant nombre de fonctions externes par du logiciel .

LE DETERMINISME DANS LES APPLICATIONS TEMPS-REEL.

Une dimension essentielle des performances d'un système temps-réel reste le déterminisme à l'exécution. Ceci est fondamental au niveau de la durée de développement du système ainsi que lors de son comportement face aux pires conditions dans son environnement final. Le déterminisme à l'exécution permet au programmeur de connaître le nombre exact de cycles machine nécessaires au processeur pour exécuter une partie donnée de l'application.

Le processeur souffre d'un certain indéterminisme à l'exécution si le nombre de cycles dépend de la succession des instructions, du type de celles-ci, ou encore de la valeur

des données. Par exemple, variation du temps d'exécution en fonction de la présence ou de l'absence des données nécessaires dans la mémoire cache. Si le nombre de cycles machine nécessaires pour multiplier deux entiers dépend de l'un des opérandes, il s'agit alors d'indéterminisme issu des données. Aucun de ces types d'indéterminisme n'est acceptable dans un système temps-réel.

En dernière analyse, le concepteur doit estimer les performances du système en se référant aux temps d'exécution les plus pessimistes.

Le haut niveau d'intégration actuel des architectures CISC et RISC, associé à la complexité du code exécutable produit par les compilateurs optimiseurs, empêchent les concepteurs d'apprécier correctement les temps d'exécution dans les conditions d'utilisation critiques. La plupart des options d'exécution sont masquées, soit par le contenu de la mémoire cache et la recopie des registres, soit par des situations de branchement retardé.

UNE ALTERNATIVE AUX MICROPROCESSEURS CONVENTIONNELS POUR LE TEMPS-REEL

La plupart des microprocesseurs sont optimisés pour les applications bureautiques tels que tableurs, bases de données, traitement de texte, etc... ainsi que pour les environnements station de travail. Des composants complexes sont utilisés dans ces applications pour gérer la mémoire et les données. A l'opposé, les microcontrôleurs profitent de la croissance de l'intégration, rendue possible par les progrès technologiques, afin d'offrir des produits dédiés à ces applications. Une grande variété de circuits destinés à des applications spécifiques, ainsi que des produits d'usage plus général, ont fait leur apparition. Pourtant, les microcontrôleurs ont toujours été plus lents que les microprocesseurs, et n'ont actuellement pas été optimisés pour le temps-réel.

Les microcontrôleurs peuvent être globalement classifiés comme génériques ou spécifiques à l'application. Les produits génériques, cas du RTX2000, offrent des solutions pour un grand éventail d'applications. Le RTX2000, grâce à son bus ASIC novateur, permet l'optimisation de la solution en simplifiant le partage entre HARDWARE et SOFTWARE pour l'utilisation de périphériques externes. L'extension évidente de cette philosophie est d'intégrer ces derniers. Des produits spécifiques peuvent ainsi être développés comme une extension de l'architecture de base du processeur.

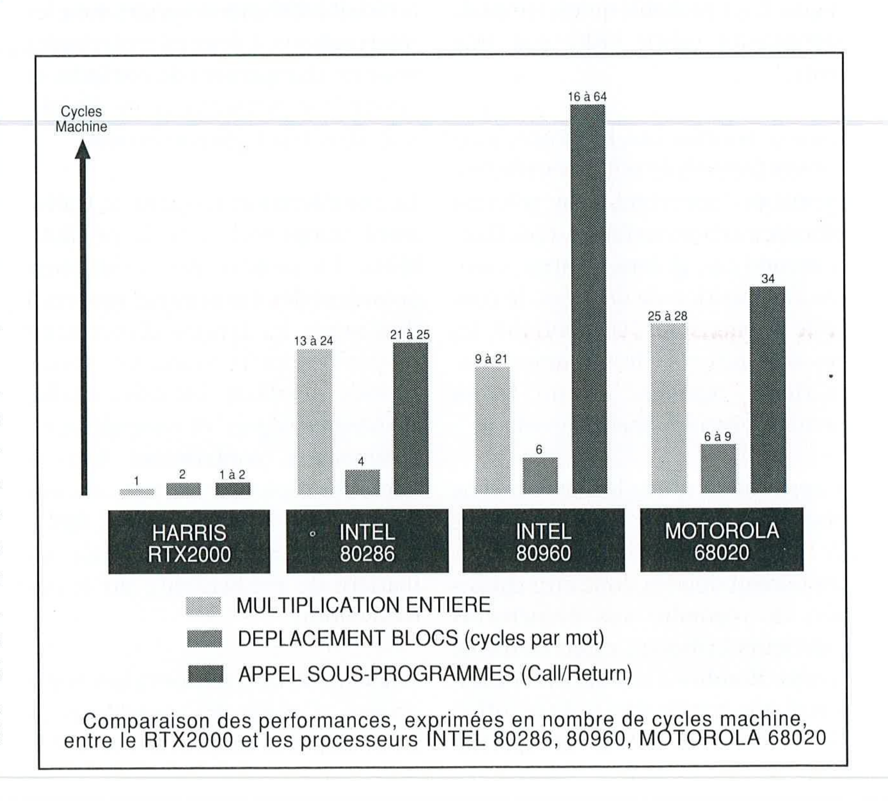
L'apparition de microcontrôleurs 16 bits rapides donne aujourd'hui la possibilité de résoudre par logiciel des problèmes confiés auparavant à des circuits externes. La limitation principale de cette migration réside dans les contraintes de performances des solutions logicielles. Le gain substantiel de puissance apporté par le RTX2000 autorise le remplacement de boîtiers logiques externes par du logiciel, améliorant ainsi la souplesse et diminuant d'autant la durée du cycle études-mise en vente. Par exemple, la rapidité du RTX2000 permet le développement d'un UART full-duplex émulé par logiciel et n'occupe qu'un pour cent de la bande passante du processeur. Cet UART logiciel remplit les mêmes fonctions qu'un UART matériel constitué de 1500 portes logiques.

RTX2000: LA PERFORMANCE AU TRAVERS DE LA SIMPLICITE

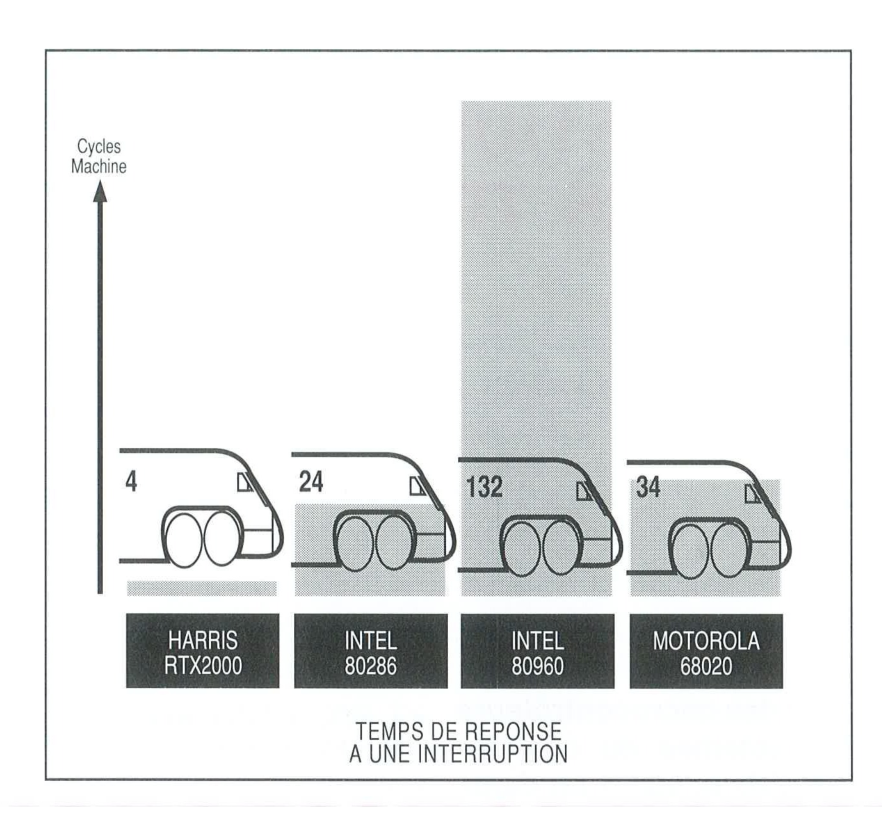
L'architecture du RTX partage certaines de ses meilleures caractéristiques avec les processeurs RISC classiques. Elle suit en particulier le modèle RISC d'opérations de chargement/mémorisation, en restant très près de la bande passante du bus mémoire.

Comme sur les processeurs RISC, les références d'opérandes se limitent à la seule mémoire présente sur la puce.

En revanche, le RTX va plus loin dans ce domaine, car, à l'exception d'un petit nombre d'instructions, il n'est nécessaire de spécifier ni l'adresse des données, ni celle de leur source ou de leur destination.



Comparaison des performances, exprimées en nombre de cycles machine, entre le RTX2000 et les processeurs INTEL 80286, 80960, MOTOROLA 68020



trôle temps-réel. Il ne supporte pas le code réentrant: si 16 utilisateurs exploitent l'éditeur, il chargera celui-ci 16 fois en mémoire... De plus, il ne possède que des mécanismes simples pour la synchronisation des communications entre tâches.

Un des avantages d'une machine orientée pile réside dans sa capacité à gérer la réentrance. Cette réentrance est utile dans les systèmes temps-réel:
- optimisation de la mémoire: plusieurs tâches peuvent utiliser le même code,
- adapté à l'environnement multitâches: les interruptions peuvent également utiliser ce code après un changement de contexte.

FORTH produit un code qui, par essence, convient à la réentrance.

Les microcontrôleurs de la famille RTX, comme les processeurs RISC, exécutent leurs instructions à l'intérieur d'un seul cycle machine, sauf pour les instructions de référence à la mémoire s'exécutant en deux cycles machine. C'est également dans le silicium que se situe le décodage des instructions du RTX2000. Son jeu d'instructions est codé horizontalement en champs dédiés, avec des fonctions spécifiques d'exécution simultanée. Le RTX2000 possède le haut niveau de parallélisme caractérisant la plupart des RISC; ainsi jusqu'à quatre de ses bus internes peuvent être simultanément actifs pendant un seul cycle machine.

LA COMPLEXITE DU DEVELOPPEMENT EN ENVIRONNEMENT TEMPS-REEL

Les différents facteurs caractérisant les logiciels pour systèmes temps-réel par rapport aux applications conventionnelles de traitement des données, sont:
- la réponse effective aux *stimuli* du monde extérieur,
- la réponse en un temps déterminé,
- la manipulation directe des ressources matérielles,
- la synchronisation des processus,
- le haut degré de fiabilité.

La famille RTX dispose d'outils de développement logiciel et d'une architecture rapide pour opérer dans l'environnement temps-réel. Elle permet un environnement de travail interactif fondé sur quatre attributs primaires:
- jeu d'outils hautement intégrés,
- faible coût et facilité d'utilisation,
- allègement de l'application,
- exploitation de la structure intrinsèque du composant comme outil.

LES PROGRAMMES REENTRANTS EN ENVIRONNEMENT MULTI-TACHES

Bien que tous les systèmes d'exploitation temps-réel soient multitâches, la réciproque n'est pas vraie. UNIX, par exemple, est trop lent pour répondre aux interruptions et effectuer un changement de contexte adapté aux applications temps-réel. Sa structure interne convient au développement logiciel, non au con-

APPLICATIONS DES MICROCONTROLEURS RTX2000

Les microprocesseurs RTX ont été incorporés dans un grand nombre d'applications d'origine très différente:
- compression de la voix (utilisation de l'algorithme CVSD).
- équipement de contrôle audio professionnel par télécommande.
- reconnaissance de la voix pour identification,
- tomographie industrielle et médicale,
- reconnaissance de forme pour télésurveillance,
- unité centrale d'une interface SCSI.

Ces produits sont également bien adaptés à d'autres applications, telles que les compteurs électroniques, contrôle de mouvement, encryptage et décryptage des données, systèmes de navigation embarqués, et plus généralement tout ce qui touche à l'acquisition numérique rapide.

CARACTERISTIQUES ET ARCHITECTURES TEMPS-REEL

RTX 2000 - RTX2001A - RTX 2010

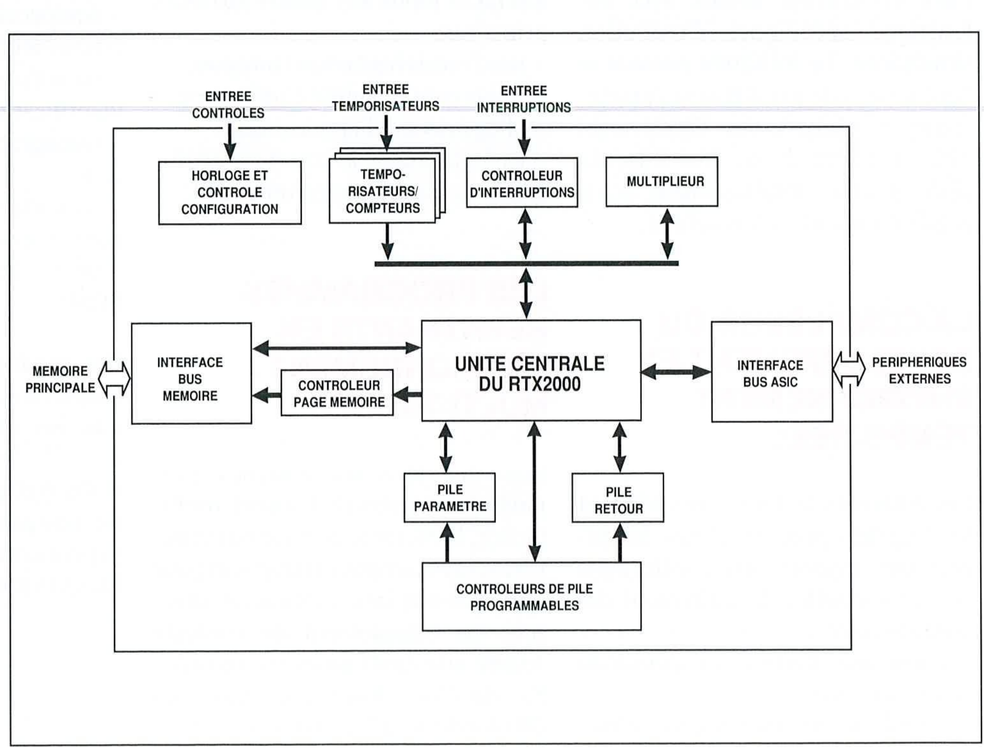
Les circuits RTX de HARRIS sont des microcontrôleurs CMOS 16 bits destinés aux systèmes de contrôle temps-réel, consommant 5mA/MHz et 500 A en mode standby.

L'architecture RISC du RTX apporte une très grande vitesse d'exécution du contrôle de flux de données:

- quatre bus orientés autour de deux piles, la première contenant les paramètres et destinée au passage de données entre sous-programmes, la seconde gérant l'adresse de retour de sous-programme,
- instructions exécutées en un cycle machine, à l'exception de celles faisant référence à la mémoire de donnée externe,
- pas d'antémémoire, ni de pipeline , donc prédictibilité à l'exécution du code,
- temps de réponse rapide et fixe de 400 ns aux évènements externes,
- flexibilité par la visibilité des registres via le bus d'entrées/sortie,
- périphériques intégrés, un multiplieur 16 bits, un contrôleur d'interruptions, trois compteurs/temporisateurs 16-bits et deux contrôleurs de piles.
- espace utilisateur pour les changement de contexte en environnement multi-tâches.

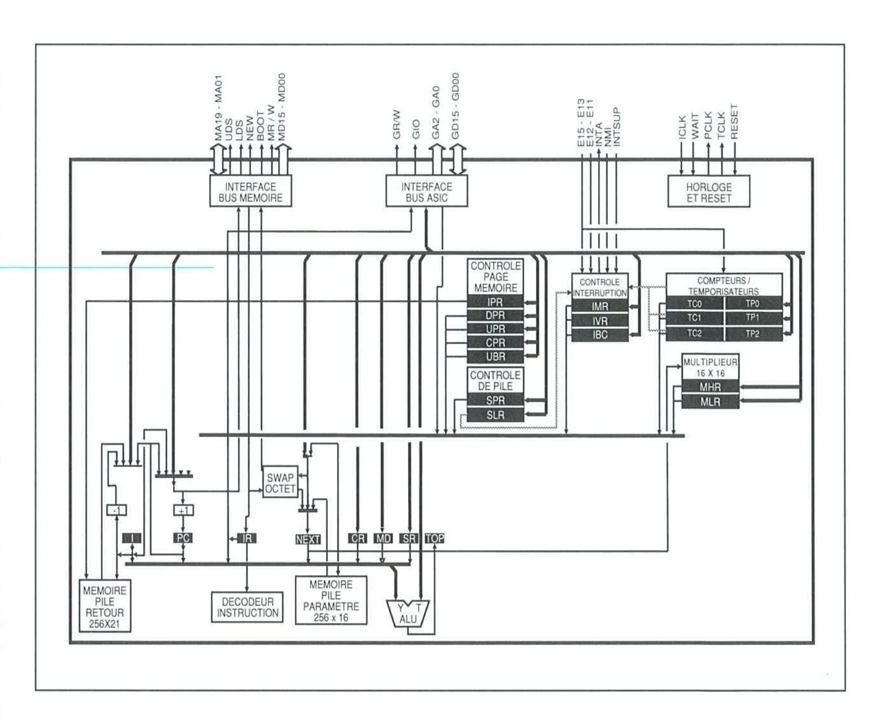
L'architecture du RTX a été conçue à l'aide de l'atelier logiciel *RTX Toolbox,* développé pour le programme RTX. Celui-ci comprend un ensemble complet d'outils de conception et de briques réalisé à partir d'une bibliothèque de cellules standard. Grâce à elles, les concepteurs bénéficient en les réutilisant d'un très grand degré de souplesse et d'automatisation.

Le RTX a été conçu pour exécuter directement les instructions en langage évolué Forth, supprimant ainsi tout recours à la programmation en assembleur. Le jeu d'instructions



microcâblé apporte les fonctions nécessaires pour coder la plupart des instructions Forth directement dans le coeur du RTX. Ce dernier dispose d'instructions spécifiques pour manipuler les piles, accéder à la mémoire, contrôler l'exécution de programme, opérer des calculs arithmétiques et logiques.

Une instruction du RTX peut combiner jusqu'à 4 instructions Forth évoluées, portant la vitesse d'exécution à un débit supérieur à celui de l'horloge du processeur, soit 15 à 40 millions d'opérations par seconde pour une vitesse d'horloge de 10 MHz. Avec son architecture en pile DUAL STACK, le RTX est tout particulièrement conçu pour exécuter plusieurs autres langages classiques, tels que le C, orienté autour du débogueur, et ADA.



RTX: UNE ARCHITECTURE QUATRE BUS PARALELLES ORIENTEE PILE

Le RTX est un microcontrôleur disposant de deux piles intégrées. Toutes les fonctions mathématiques, les entrées/sorties et les accès mémoire reçoivent et transmettent leurs paramètres respectivement de et vers la pile paramètre. Les appels de procédures utilisent la pile de retour.

Le RTX contient 23 registres 16-bits. Ces registres contrôlent la configuration et l'état du microcontrôleur, conservent les résultats intermédiaires et servent d'interface avec les périphériques intégrés. Les registres et les piles sont tous reliés par des bus 16-bits chargés du transfert de données entre le processeur et son environnement.

LES PILES DUAL STACK

Les deux piles de type LIFO du RTX ont chacune 256 niveaux. L'élément supérieur de chaque pile est accessible directement par un registre. Le reste de la pile est contenu dans une mémoire RAM intégrée. La logique de contrôle associée à chaque pile détermine quel emplacement dans la pile doit être lu ou écrit.

La pile paramètre reçoit les opérandes traités par les opérations mathématiques, logiques et d'accès mémoire, ainsi que le passage de paramètres entre sous-programmes. Une opération d'empilage ou de dépilage est exécutée en un seul cycle machine.

La pile de retour mémorise les adresses de retour des sous-programmes et peut contenir un index de boucle lors de l'exécution des instructions itératives. La pile de retour peut aussi être utilisée comme zone de stockage temporaire.

UNE ARCHITECTURE QUATRE BUS INTERNES

L'architecture en QUAD BUS consiste en 4 bus 16-bits indépendants, actifs simultanément:

- le bus de mémoire principale, composé de deux bus distincts, bus d'adresses et bus de données, chargé du transfert des instructions et données du programme. Un échangeur d'octets contrôle l'ordre de lecture/écriture dans la mémoire au format INTEL ou MOTOROLA. Grâce au bus 20 bits (UNS/LAS) destiné aux accès mémoire, le RTX adresse directement 512 KMots de mémoire, soit 16 pages de 32 KMots.
- le bus 16 bits de la pile paramètre empile le résultat des opérations provenant de l'ULA, la mémoire principale ou des entrées/sorties.
- le bus 21 bits de la pile de retour empile l'adresse du retour de sous-programme.
- le bus rapide destiné aux entrées/sorties du RTX, appelé bus ASIC, est le bus principal de commande. Il accède directement aux registres internes et aux périphériques externes. Il est possible de connecter au bus ASIC des sources externes de données rapides ou des accélérateurs de traitement servant d'extensions naturelles à l'architecture interne. Le taux de transfert est de 20 Moctets par seconde. De plus, le bus ASIC donne accès en lecture et écriture, en un cycle machine, aux registres internes.

LES REGISTRES

Le RTX contient trois sortes de registres, les registres relatifs aux piles, les registres de statut/contrôle accessibles à travers le bus ASIC, et divers autres registres qui ne sont pas directement accessibles:

- registre de sommet de pile (T pour Top). Il contient le sommet de la pile paramètre. Toutes les fonctions de l'ALU sont alimentées par T. Le con-

tenu de T peut être dirigé vers tous les registres du bus ASIC et les entrées/sorties.
- registre suivant (N pour Next). Il contient le second élément de la pile paramètre. N sert de pointeur pour toutes les opérations d'écriture/lecture en mémoire.
- registre d'index (I pour Index). Il contient les 16 bits de poids faible du sommet de la pile de retour, auxquels sont associés les 5 bits du registre IPR (IPR pour Index Page Register).
- registres d'état ou de contrôle du RTX. Ils déterminent l'environnement et permettent au processeur de contrôler et d'agir sur les E/S.
- registre d'instruction (IR). Il contient l'instruction en cours d'exécution. Les bits de cette instruction sont décodés pour déterminer quelle opération interne doit être effectuée, l'emplacement de la prochaine instruction à exécuter, et pour apporter une donnée immédiate. IR est chargé directement depuis la mémoire centrale et n'est pas accessible par programme.

Tous ces registres, excepté IR, sont accessibles depuis le bus ASIC. Chaque registre possède une adresse bien définie. Le RTX peut adresser 32 périphériques ASIC. Les adresses ASIC 0 à 23 (17H) sont réservées pour les registres et les E/S intégrées.

LA MEMOIRE

Dans le RTX2000, on trouve trois sortes d'instructions de référence à la mémoire: accès en mémoire programme, accès en mémoire données et accès en mémoire utilisateur. Le jeu d'instructions du RTX dispose de classes d'instructions se référant à chacun de ces espaces mémoire:
- mémoire programme. Elle contient les instructions exécutées par le RTX. Le bus adresse mémoire de 20 bits du RTX est décomposé en 16 bits d'adresse provenant du registre T et 4 bits du registre de page.
- mémoire données. La classe d'instructions se référant à la mémoire données comprend les adresses RAM utilisées pour les variables et le stockage des données. Le RTX y accède par 16 ou 8 bits. Les accès sont considérés sur 8 bits lorsque l'octet de poids fort de T est à zéro.
- mémoire utilisateur. L'espace mémoire utilisateur est un bloc de 32 mots accessibles pour le RTX sans avoir à charger d'adresse sur la pile de retour. L'adresse logique à référencer dans le bloc est comprise dans le code de l'instruction. Cette zone de mémoire sert au stockage des données nécessitant un accès fréquent, cas des paramètres système ou des sauvegardes de contexte de tâche en mode multi-tâches.

APPEL RETOUR DE SOUS-PROGRAMME EN UN SEUL CYCLE MACHINE

Quand un appel de sous-programme est exécuté, l'adresse de l'instruction suivante est sauvegardée dans la pile de retour. En fin de sous-programme, l'exécution se poursuit à l'adresse sauvegardée sur la pile de retour. Si le retour de sous-programme est intégré au code de l'instruction en cours de traitement par mise à un bit de retour (return bit), celui-ci sera exécuté en zéro cycle machine et en un cycle dans les autres cas.

BRANCHEMENTS ET BOUCLES

Le RTX exécute des branchements inconditionnels ou conditionnels en fonction du contenu des registres T ou I.

Tous les branchements s'exécutent en un cycle, qu'il soit effectif ou non. Pour les boucles rapides, on peut faire appel à la forme NEXT du branchement conditionnel. Dans ce cas, le nombre d'itérations est chargé dans le registre I.

LES INSTRUCTIONS ITERATIVES

Les itérations peuvent exécuter un bloc d'instructions sans recourir à un cycle recherche et décodage, option fréquemment utilisée pour les transferts de données, les boucles ou certaines opérations mathématiques.

LES OPERATIONS DE L'UNITE ARITHMETIQUE ET LOGIQUE

L'ULA prend en charge les opérations d'addition, soustraction et les opérations logiques AND, OR, XOR, NOR, NAND et XNOR. Le registre T correspond toujours à une des entrées de l'ULA. La seconde entrée provient des registres ou des entrées/sorties. Le contenu des registres T et N est toujours utilisable comme opérandes de l'ULA.

LES INTERRUPTIONS

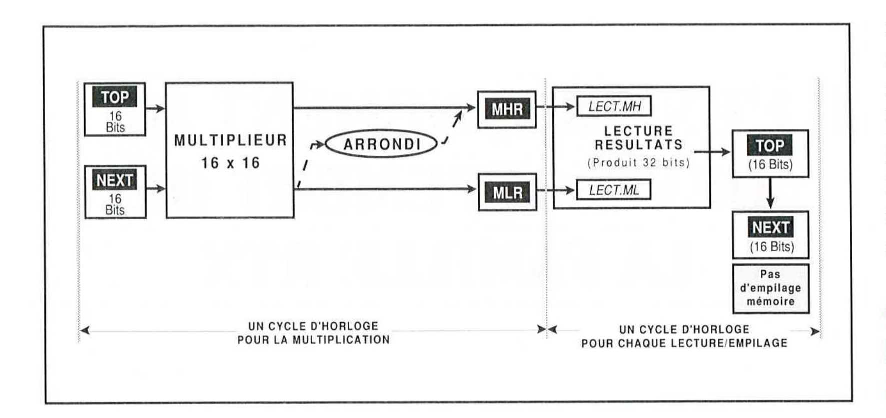
Le RTX peut être interrompu par des sources internes ou externes. Le contrôleur d'interruptions intégré accepte 14 entrées dont 13 masquables et une non masquable. Il vérifie ces entrées à chaque instruction et rend prioritaire une demande active, puis la signale au contrôleur. Pendant ce cycle, le processeur sauvegarde l'adresse d'exécution courante sur la pile de retour, inhibe les interruptions, puis lit un vecteur pointant sur l'adresse du sous-programme de traitement de l'interruption correspondante.

LE MULTIPLIEUR (RTX2000) ET LES OPTIONS VIRGULE FLOTTANTE (RTX2010)

L'unité de calcul dispose d'un multiplieur parallèle 16x16 bits intégré. Le multiplieur travaille selon deux modes de précision distincts:
- produit complet sur 32 bits,
- ou 16 bits significatifs avec ou sans arrondi.

L'option d'arrondi est programmée avec un seul bit de configuration. Les opérandes de la multiplication proviennent des registres T et N. Le résultat est stocké dans T et N, ces deux registres étant considérés dans ce cas comme registre 32 bits.

Trois commandes distinctes contrôlent la multiplication:
- avec signe, en complément à deux,



- sans signe,
- suite de produits. Dans ce cas, le premier opérande est fixe et se trouve dans le registre N, tandis que la succession des autres opérandes passe de la mémoire vers le multiplieur à travers le registre TOP. Les bits de poids fort du produit arrondi sont alors disponibles. Cette opération est particulièrement utile pour le décalage ou le changement d'échelle de gros blocs de données ou d'adresses, dans les traitements graphiques par exemple.

Le RTX2001A, produit faible coût dérivé du RTX2000, ne dispose pas du multiplieur intégré. Il conserve cependant tous les registres 16 bits pour les opérations racine carrée, division, afin de conserver une portabilité montante.

Le RTX2001A possède deux piles intégrées de 64 niveaux et présente la plupart des caractéristiques du RTX-2000 mais il dispose d'un contrôleur d'interruption renforcé pour les opérations multi-tâches.

Le RTX2010, compatible broche à broche avec le RTX2000, possède une unité mathématique à virgule flottante.

CONTROLEUR DE PILES

Le RTX-2000 possède deux contrôleurs de pile programmables pour assurer l'adressage implicite des deux piles et surveiller les positions des pointeurs de pile dans chaque espace-adresse de celle-ci. Chacun d'eux peut placer ou retirer jusqu'à 256 mots par pile.

Chaque registre programmable LIMIT des contrôleurs de pile, fixe le nombre maximal de mots que l'on peut placer sur chacune d'elles sans engendrer de débordement. L'utilisateur peut affecter une limite de pile inférieure à sa profondeur physique pour y réserver un espace de manoeuvre. De même, une interruption se produit lors d'une tentative de dépilage de données alors que le registre T est vide. Les valeurs des deux registres 8 bits LIMIT sont contenues dans un registre 16 bits de limite de pile.

CONTROLEUR D'INTERRUPTIONS

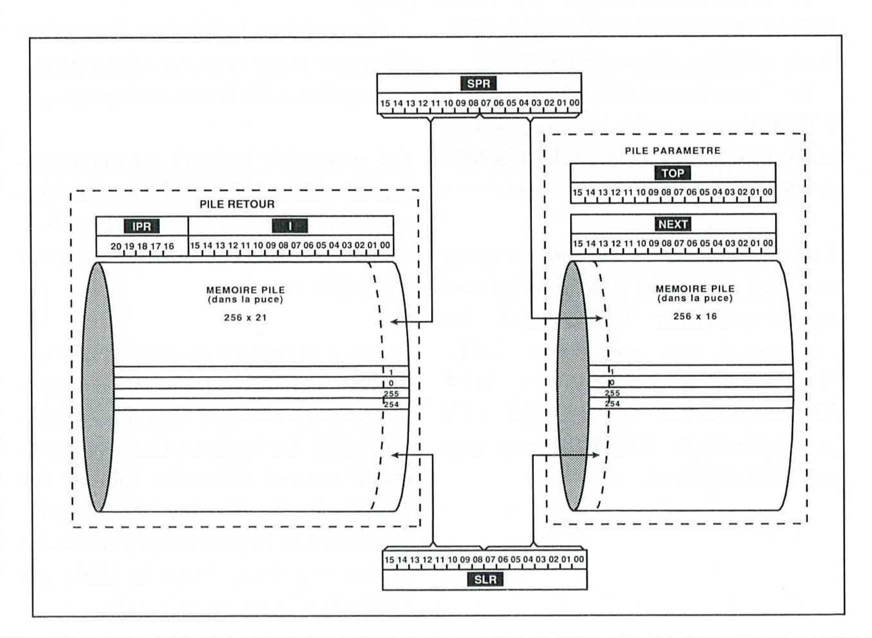
Les interruptions internes et externes sont gérées par le contrôleur d'interruptions gérant 14 niveaux de priorité. La priorité la plus élevée est attribuée à l'interruption non masquable (NMI). Neuf de ces interruptions servent à traiter les dépassements haut ou bas des piles, la NMI, l'interruption logicielle et les trois registres compteurs/temporisateurs. Les cinq autres niveaux d'interruption sont disponibles pour les broches d'interruptions externes.

COMPTEURS D'EVENEMENTS ET REGISTRES D'HORLOGE

Le RTX2000 possède trois compteurs/temporisateurs 16 bits, configurables séparément comme compteurs d'événements ou comme registres d'horloge.

Si les compteurs sont configurés en mode comptage décrémental (comptage d'événements), ils sont prêts à recevoir des signaux externes sur l'une des broches partagées avec les entrées d'interruptions. Dans ce mode, les compteurs engendrent une interruption à chaque passage au zéro.

En mode registre d'horloge, les compteurs sont alimentés par l'horloge interne du système et avancent en synchronisme avec l'horloge interne du processeur. Par défaut, tous les compteurs sont considérés comme registres d'horloge.



Chaque temporisateur/compteur comprend un registre de préchargement, un compteur 16 bits, un circuit de sélection d'horloge, et une sortie d'interruption pour synchroniser les sémaphores dans les systèmes multi-tâches temps réel.

LA GESTION MEMOIRE MULTI-TACHES

Le RTX adresse directement 512 KMots de mémoire, partagés en 16 pages de 32 KMots. L'espace code est accessible par toutes les instructions de chargement. L'espace données se réfère aux instructions de référence mémoire. L'espace utilisateur permet de disposer de façon très efficace d'un bloc de 32 mots se trouvant n'importe où en mémoire.
Pour offrir un solution complète la famille RTX comprend les périphériques suivant:
- le RTX2152 2Mbits UART. Un récepteur/transmetteur asynchrone universel programmable, de haute performance, et un générateur de vitesse de transmission qui toléreront des vitesses de transmission des données allant jusqu'à 2.06 Mbits avec une horloge 16X (0 à 33 MHz). Ce produit fournit une interface directe RTX et il est totalement opérationnel après avoir programmé seulement trois registres internes 8 bits.
- le contrôleur LAN ARINC RTX1553, codeur/décodeur manchester.
- le microcontrôleur 16 bits RTX2010 avec unité de virgule flottante intégrée, disponible en 1990,
- le microcontrôleur 32 bits RTX4000 avec unité de virgule flottante intégrée, en cours de développement.

En complément, Harris propose pour les solutions aux applications microcontrôleur temps réel, les interfaces bus industriel VME, MULTIBUS II , GESPAC , MCA NUBUS et SUNBUS de la famille FCT en technologie BICMOS avec une sortance de 64 mA.

□

L'ENVIRONNEMENT DE DEVELOPPEMENT DE LA FAMILLE RTX

La famille RTX2000, RTX2001A, RTX2010 dispose d'un environnement de développement matériel et logiciel intégré.

Le système de développement RTXDS constitue la partie matérielle et est relié à un système hôte, un micro-ordinateur de type IBM PC/XT/AT ou compatible, par une liaison série.

Le système hôte supporte la version PC-FORTH comprenant un éditeur d'écrans, un compilateur FORTH compatible 8088/286, un compilateur cible pour le RTX2000, un désassembleur et un moniteur de mise au point.

L'ensemble logiciel comprend:
- un émulateur logiciel qui télécharge le programme développé depuis le micro-ordinateur hôte vers le système de développement via la liaison série,
- des routines logicielles, tels qu'un désassembleur et un moniteur de mise au point associant le code source.

Cet ensemble logiciel est accompagné de manuels de références techniques, Harris RTS Software Development System, en langue française sur demande.

Grâce à la nature structurée et modulaire du langage Forth, le programme est mis au point et testé sans difficulté. Le système de développement permet d'émuler, placer des points d'arrêts, désassembler le code, visualiser le contenu de la mémoire et des registres, lister la table des symboles et imprimer le code.

L'ENVIRONNEMENT LOGICIEL

Le langage FORTH est particulièrement adapté au développement des applications temps réel. Avec FORTH, écrire une application consiste à définir de nouveaux mots ou instructions de manière interactive. Ces mots, similaires aux procédures définies dans des langages plus conventionnels, sont exécutables en mode interprété immédiatement après compilation, que ce soit depuis le clavier ou depuis un fichier source.

Les mots définis par l'utilisateur deviennent partie intégrante du dictionnaire. Ils peuvent utiliser toutes les primitives pré-définies ou devenir eux-mêmes primitives des mots à définir. Si un mot est intégré à la définition d'une procédure plus générale, il reste cependant exécutable individuellement. Cette interactivité est un des atouts majeurs du langage de développement Forth. Ceci permet une définition des tâches exécutées par la machine en éléments faciles à contrôler, tout en conservant une structuration du programme global.

En plus des types de données prédéfinis, Forth peut créer de nouveaux mots de définition, technique similaire à celle des langages orientés objet. Forth permet la création d'un répertoire virtuellement illimité de types de données et de structures.

Une grande partie du programme utilisateur peut être développé de façon autonome et débogué uniquement dans l'environnement hôte. Après la compilation croisée, les mots spécifiques à la machine, habituellement liés aux matériels et/ou critiques en temps d'exécution, sont implantés dans le système cible. Le code résultant est ensuite transféré dans la mémoire locale du RTX2000 pour exécution et test.

Le désassembleur permet de visualiser le code objet généré. Aucune optimisation n'est requise, la structure du programme source étant en parfaite concordance avec le code objet.

Le compilateur cible produit un code binaire pur, dépourvu d'en-tête, directement implantable en mémoire morte, ce sans recourir à un noyau exécutable ou à un système d'exploitation temps-réel externe.

Hormis Forth, le langage C est disponible grâce à un compilateur croisé C fonctionnant sur PC. Un débogueur en ligne permet un développement rapide dans un environnement à fenêtres. Un noyau multi-tâches, profitant des spécificités de l'architecture de la famille RTX, sera prochainement disponible.

Le cycle normal de développement, ne passant plus par la traditionnelle phase ECLR (Edit Compile Link and Run), est ainsi simplifié d'autant.

La plupart des programmes FORTH non spécifiques à une machine sont développés et mis au point en environnement hôte. Après une première compilation croisée, les mots spécifiques à la machine sont traduits en code cible. Ce code cible est ensuite téléchargé dans la mémoire du RTX-2000 pour exécution.

Le système de supervision hôte-cible du RTXDS assure la surveillance permanente de l'exécution du programme en mode pas à pas ou en continu.

Les références de mise au point sont en clair. Il ne s'agit donc pas de simples références de pointeurs.

Le désassembleur renvoie le code compilé sous forme d'une suite d'instructions machine exprimées en primitives RTX2xxx FORTH. Le flux d'exécution peut être vérifié et permet le comptage des cycles machine.

MATERIELS DE DEVELOPPEMENT RTX

Le système de développement RTXDS contenant:
- le microcontrôleur RTX2000,
- 16 Ko octets d'EPROM,
- 32 Ko octets de mémoire RAM,
- ports d'entrée parallèles,
- trois ports de sortie parallèles,
- un UART
- et un emplacement dédié aux développements de l'utilisateur (extensions mémoire sur demande).

L'émulation logicielle du RTX offre des possibilités de développement simultanées. La phase de contrôle et de mise au point nécessitera la présence effective du système de développement.

Les développements destinés au RTX2010, version à virgule flottante, peuvent s'effectuer sur le RTXDS.

La carte de développement RTXDB-10 ou RTXDB-8 constituent un système d'évaluation à architecture minimale. Elle permet le développement d'applications à partir des logiciels définis par l'utilisateur ou grâce au logiciel de développement RTS Harris acquis séparément.

Kits d'extension mémoire pour RTXDS:
- référence RTX-MX64, 64 Ko de SRAM à 35 ns de temps d'accès. Un emplacement pour extension mémoire de 128 Ko est prévu.
- référence RTX-MX192, 192 Ko de SRAM à 35 ns de temps d'accès.

Les produits RTX présentent des performances tout à fait remarquables pour les applications temps réel exécutant des calculs ultra-rapides sur des entiers ou incluant des processus de décision tels les tris, recherches et comparaison de structures.

La simplicité matérielle de ses interruptions et de ses appels de procédures le destinent aux applications où la prévisibilité d'exécution et la connaissance précise du nombre de cycles machine sont indispensables.

Grâce à la structure matérielle et logicielle des circuits RTX, la mise au point finale peut être réalisée sur l'application elle-même sans l'aide de l'émulateur, facilitant ainsi l'intégration sur le site.

LANGAGE C

Les applications temps-réel tendent vers une complexité supérieure aux applications standards, au niveau logiciel, et souffrent paradoxalement d'un manque de caractéristiques adéquates dans les langages de haut niveau. En effet, la facilité et la vitesse de développement deviennent des critères de plus en plus importants dans le choix d'une architecture à base de microcontrôleur.

Grâce au langage FORTH couvrant à la fois les besoins de rapidité temps réel et offrant les structures d'un langage de haut niveau, le programmeur dispose ainsi d'un environnement logiciel C orienté FORTH très performant.

Le compilateur RTS-C, développé pour la série des microcontrôleurs RTX, est un compilateur croisé générant du code FORTH très efficace. Le compilateur est disponible pour les développeurs de système temps réel préférant programmer en langage C, et profiter ainsi des librairies standards et des librairies spécifiques aux composants des séries RTX2000 (multiplieur intégré) ,RTX2001A (sans multiplieur) et RTX2010 (avec virgule flottante).

Le compilateur croisé C, fonctionnant sur PC, associé à un débogueur au niveau ligne C, permet un développement rapide dans un environnement à fenêtres.

Ce produit compile le code évolué du langage C en code FORTH exécutable par le RTX, et permet d'inclure

```
Step Display Break Watch View Assign Find Go Options Quit
                         Break
 1: #inc At line
 2:      At Cursor Position              Ctrl-F8
 3: #def At Function Entry
 4:      Show Breaks                     F5
 5:main  Delete Breaks
 6:{     Remove All Breaks               Ctrl-F5
 7:  in
 8:  int array[SIZE];
 9:  int total = 0;
10:  index = 0;
11:  do
12:  {
13:    array[index] = index;
14:    total += array[index++];
15:  }
16:  while (index < SIZE);
                    Output Window
Program running ....

File:sample.c Line: 9 Context: Waiting for a command
```

également des primitives FORTH. L'environnement du compilateur intègre l'affichage du nombre de cycle machine pour chaque instruction en langage évolué C. Grâce au débogueur, le développeur visualise le code objet en correspondance avec les lignes du programme source.

La possibilité d'associer le programme et le nombre de cycles machine nécessaires à son exécution représente un atout majeur dans les applications temps-réel où le temps d'exécution est un facteur critique. Le compilateur produit un code exécutable binaire directement implantable en mémoire morte. Ce code ne requiert ni noyau exécutif ni système d'exploitation temps-réel externe.

Le compilateur contient un ensemble d'outils de programmation:
- un environnement multi-fenêtres,
- une mémoire de trace,
- un accès aux commandes MS-DOS,
- un utilitaire de création,
- des directives Forth en ligne,
- un débogueur symbolique des instructions du langage C,
- une bibliothèque standard,
- des primitives dédiées à l'architecture RTX.

L'outil de mise au point supporte les interruptions, affiche le contenu des variables, indique les points d'arrêt. Avec les options *instruction*, *affichage*, *interruption*, *surveillance*, *visualisation*, *affectation* et *localisation*, l'utilisateur peut ainsi développer les applications RTX en disposant de toutes les facilités de mise au point. □

TURBO - FORTH

OFFREZ-VOUS UN EXCES DE VITESSE

A l'encontre des avis autorisés, ou censés l'être, FORTH n'est ni une religion, ni une chapelle. C'est un langage de programmation, véritable environnement de développement adaptable à toutes les situations.

UN LANGAGE A PART ENTIERE

FORTH exploite un ensemble de procédures pré-définies incorporées à un dictionnaire. Ces procédures échangent leurs données en les faisant transiter par une zone mémoire particulière, la pile de données. Cette particularité, qui n'est pas spécifique à FORTH, impose l'utilisation de la notation postfixée pour l'ensemble des opérateurs.

Le jeu de primitives, très riche, est extensible par l'utilisateur en définissant de nouvelles procédures appelés aussi mots, des variables, des constantes, et de nouveaux mots de définition. FORTH fait appel à tous les types de structures de contrôle, boucles itératives et récursivité.

UNE ETRANGE DUALITE

FORTH est à la fois interpréteur et compilateur. Tout mot tapé au clavier est immédiatement interprété. La compilation n'est qu'un état particulier de l'interpréteur.

FORTH compile ou interprète indifféremment un flot d'instructions provenant du clavier ou d'un fichier source. Dans un programme, tout mot défini peut être immédiatement compilé ou exécuté.

Le compilateur FORTH ne gère aucune table de référence, bibliothèque ou module *run-time*. Les liens avec les procédures pré-définies sont créés pendant la compilation. Le programme source est traité en une seule passe.

Le programme est exécutable immédiatement. Dans la phase de mise au point, chaque procédure est exécutable individuellement. Le programmeur a accès à toutes les données, variables, constantes et adresses mémoire.

La rapidité d'exécution des programmes FORTH est comparable à celle

des programmes générés à partir des meilleurs compilateurs.

LA VERSION TURBO-FORTH 83-STANDARD

Le langage FORTH souffrait jusqu'alors de particularités rendant son apprentissage peu engageant pour les débutants: fichiers en blocs, éditeur de source en ligne, environnement mal intégré aux fonctions DOS.

TURBO-Forth compile et édite des fichiers source en ASCII. Un éditeur pleine page dispose de toutes les fonctions de mise en forme de texte: recherche, remplacement, déplacement, suppression, importation. Si l'éditeur fourni avec le produit ne convient pas, l'utilisateur peut en changer, TURBO-Forth acceptant tout type d'éditeur, du simple EDLIN fourni avec DOS au plus sophistiqué des traitements de texte.

La compilation ou l'exécution du contenu d'un fichier source est lancée en tapant la directive INCLUDE suivie du nom de fichier. Cette directive peut également être intégrée au programme source pour traiter le contenu d'un autre programme source. TURBO-Forth peut imbriquer jusqu'à huit directives INCLUDE et en enchaîner un nombre infini. TURBO-Forth autorise la modularité en programmation.

Si d'autres langages acceptent l'homonymie des constantes et des variables, à la condition de les affecter à des procédures distinctes, TURBO-Forth va plus loin en tolérant la compilation des procédures ayant le même nom.

DES VECTEURS

TURBO-Forth exploite largement le concept de vecteurs. Cette technique particulière de programmation exécute une définition par double indirection du pointeur d'interprétation.

La vectorisation permet de rediriger les flux d'entrées et sorties système, l'interpréteur externe, le décompilateur. L'utilisateur peut modifier aisément l'ensemble du comportement du langage et par conséquent, celui de son application.

UN ASSEMBLEUR

TURBO-Forth dispose d'un assembleur très complet. Un mot défini en code machine est intégré au dictionnaire au même titre qu'une définition en langage évolué.

L'assembleur utilise ses propres structures de contrôle, évitant ainsi le recours abusif aux labels et aux branchements toujours lourds à gérer.

UN META-COMPILATEUR

Derrière ce terme se cache le concept le plus puissant lié au langage FORTH.

La version TURBO-Forth n'a pas été développée à partir d'un quelconque assembleur ou compilateur, mais avec TURBO-Forth lui-même.

Pour lancer une session de méta-compilation, TURBO-Forth charge d'abord un méta-générateur. Ensuite, le méta-générateur va chercher les fichiers source écrits en FORTH, et génère le code exécutable dans un segment mémoire adjacent. En fin de méta-compilation, ce code est sauvegardé sur disque. L'utilisateur dispose d'une nouvelle version de TURBO-Forth.

Si la directive de compactage est active au lancement de la méta-compilation, le programme résultant sera réduit aux seules routines effectivement utilisées par le programme d'application. La compacité du code généré est telle qu'un clone du traitement de texte WordPerfect (limité à l'édition des fichiers ASCII) tient en 16 Ko et dispose quand même de ressources avancées: recherche avant et arrière, substitution, déplacement de blocs, gestion simultanée de deux documents, définition de macros...

En chargeant un assembleur, défini dans un fichier source, destiné à des processeurs 8051, 68HC11, HARRIS RTX 2000, etc..., TURBO-Forth peut méta-générer du code exécutable sur des systèmes ne disposant d'aucune ressource matérielle et logicielle pour héberger des outils de développement.

L'INTERACTIVITE

TURBO-Forth dispose de nombreuses options paramétrables pour assister le programmeur pendant la phase de mise au point:
- l'option ASSIST? ON affiche en permanence le nombre de paramètres déposés sur la pile et la base numérique courante,
- l'option ECHO ON ou OFF active ou désactive l'affichage du programme en cours de compilation,
- l'option CAPS ON ou OFF, accepte ou non la différence entre caractères majuscules et minuscules dans le flot d'entrée,
- l'option SEE, décompile un mot. Si le désassembleur est chargé, SEE désassemble également les mots définis en code machine. SEE agit sur tout le dictionnaire FORTH, y compris les primitives pré-définies,
- l'option DEBUG, active le débogueur pour exécution en mode trace du mot à contrôler. Le mot pointé par DEBUG est affiché en surintensité lors du listage du vocabulaire ou la décompilation d'un mot.
- option HELP, affiche une ou plusieurs lignes de commentaire. Ces commentaires sont définis dans des fichiers d'extension .VOC. Une routine spéciale LEARN permet à TURBO-Forth *d'apprendre* la documentation afférente à son vocabulaire.
- l'option EOF. Ce mot provoque une fin de fichier artificielle. Ce mot est généralement suivi d'une documentation très détaillée sur le programme chargé. Avec TURBO-Forth, il n'est pas nécessaire de séparer le programme source et sa documentation.

UNE BOITE A OUTILS

Alors que la majorité des outils de développement professionnels ne mettent à la disposition des développeurs que des programmes et des bibliothèques sous forme compilée ou de fichiers objet, les concepteurs

de TURBO-Forth, croyant en la notion de langage totalement ouvert à l'utilisateur, fournissent le langage avec ses sources.

Le Package TURBO-Forth contient également de nombreux programmes destinés à être exploités en l'état ou modifiés par les utilisateurs:
- gestion cartes graphiques CGA, HERCULES, EGA,
- gestion liaison série, dont primitives de gestion MINITEL,
- accès fichiers (accès séquentiel, accès direct),
- gestion fichiers au format dBASE III/III+,
- fonctions d'animation graphique 3D,
- éditeur ASCII (clone WordPerfect),
- pas moins de trois packages de traitement mathématique en virgule flottante, dont un package mathématique CORDIC (traitement numérique à précision variable),
- définition de variables locales, notation infixée, gestion chronomètre,
- désassembleur 8086, etc...

TURBO-Forth est l'outil idéal de développement pour toutes les applications professionnelles de contrôle de processus et de traitement en temps réel. Il sera également très apprécié pour le développement d'applications compactes.

PROGRAMME: CONTROLE D'INTERRUPTION

A titre d'illustration, le court programme diffusé ci-contre illustre une méthode de détournement des interruptions.

Le mot **INT@** lit un vecteur d'interruption à partir de son numéro préalablement déposé au sommet de la pile de données. En sortie, **INT@** laisse le segment et l'offset du code exécutable pointé par l'interruption.

LISTING:

```
\ Détournement des interruptions 8086
\                                     M.ZUPAN 8/88
HEX
CODE INT@ ( n - seg adr ) \ lit vecteur d'interruption
   ax pop   35 # ah mov   21 int   es push   bx push
NEXT C;

CODE INT! ( seg adr n - )
   \ écrit un vecteur d'interruption
   ax pop   dx pop   bx pop   ds push   bx ds mov
   25 # ah mov   21 int   ds pop   NEXT C;
DECIMAL

\ INT@ et INT! permettent de gérer les vecteurs
\ d'interruption 8086: ils doivent être utilisés pour
\ lire ou modifier les vecteurs des 256 interruptions
\ numérotées 00 à FF de la table située en mémoire basse
\ du système (entre 0:0 et 0:400). Il est en effet
\ fortement déconseillé d'écrire directement dans cette
\ table: une interruption pouvant toujours survenir en
\ cours d'écriture, ce qui peut avoir des conséquences
\ catastrophiques.

\ EOF \ fin fichier en l'absence de l'exemple qui suit

\  ---------------------------------------------
\  Exemple d'application au vecteur 09 du Clavier
\  ---------------------------------------------
HEX
2VARIABLE OLDINT09 \ conserve vecteur initial de Int09
9 INT@   OLDINT09 2!

LABEL NEWINT09      \ nouveau traitement de Int09
   ax push
   60 # al in
   al ah mov
   80 # al and
   0 # al cmp
   0= IF   1D # ah cmp
      0<> IF   38 # ah cmp
          0<> IF               \ routine bip
              61 # al in
              FE # al and
              02 # al xor
              61 # al out
          THEN
      THEN
   THEN
   ax pop
   FAR cs: OLDINT09 S#) jmp     \ saute enfin au
traitement int09
DECIMAL

\  mots utilisateur: BIP / NOBIP
\  -------------

: BIP   ( -)
   \ vectorise int09 sur NewInt09: clavier sonorisé
   9 INT@   OLDINT09 2!
   DSEGMENT NEWINT09   9 INT!  ;

: NOBIP   ( -)
   \ restitue ancien vecteur de int09: clavier muet
   OLDINT09 2@   9 INT!  ;

\ ATTENTION: ne pas oublier de restituer le vecteur 09
\       par NOBIP quand vous quitterez TURBO-Forth (à
\       moins de le laisser résident) sous peine de
\       plantage au premier effleurement du clavier!
```

Pour tout renseignement concernant TURBO-Forth, contacter:
REM CORP 17, rue de la Lancette 75012 PARIS, tel: 43.40.96.53 fax:43.42.32.15

RESEAUX ET CODE BARRES

L'EXEMPLE CROSS-BAR

En matière de réseau industriel codes barres, plusieurs considérations entrent en ligne de compte : rapidité des transactions, fiabilité de la transmission, possibilités de dialogue entre postes, etc...

SPECIFICITES D'UN RESEAU CODES BARRES

Le type d'informations saisies porte ici sur des quantités de données relativement faibles (10 à 200 caractères), ce à l'opposé des fichiers beaucoup plus importants échangés par les réseaux locaux de type bureautique.

La saisie codes barres est bien adaptée pour la saisie de données en milieu industriel, ce en raison de sa bonne résistance aux salissures et aux conditions difficiles. Il est donc nécessaire que le système de transmission utilisé présente les mêmes qualités de résistance à des conditions externes difficiles: parasites électriques, ambiance "sale", utilisation par un personnel non informaticien.

La sécurité des données est assurée à trois niveaux:
- à celui du terminal de saisie ou du contrôleur de transaction qui doit posséder une mémoire C-MOS ;
- à celui du concentrateur multiplexeur, qui peut posséder sa propre mémoire protégée des coupures secteur;
- enfin à celui du PC ou du mini, dans le cas où la connexion n'est pas faite en direct avec le site central..

Par ailleurs, le protocole d'échange des données entre le concentrateur multiplexeur et les terminaux assure lui aussi la sécurité des données, ce tout en limitant les échanges d'informations non-utiles. En particulier, il s'assure que les données pourront être conservées dans les terminaux jusqu'à ce que le concentrateur ou même l'unité centrale du système utilisé les ait prises en charge. Typiquement, ceci peut être assuré par la mémorisation des données par le terminal qui ne les effacera qu'après réception d'un message spécifique du concentrateur ou du système central. Les réseaux codes barres disposent d'un tel protocole, livré en standard avec les programmes internes des terminaux, imprimantes, lecteurs, et concentrateurs multiplexeurs.

Quant à l'utilisation de terminaux programmables, elle permet le contrôle de saisie, contrôle de vraisemblance, en mode local. Cette technique supprime ainsi une grande partie des échanges avec le site central.

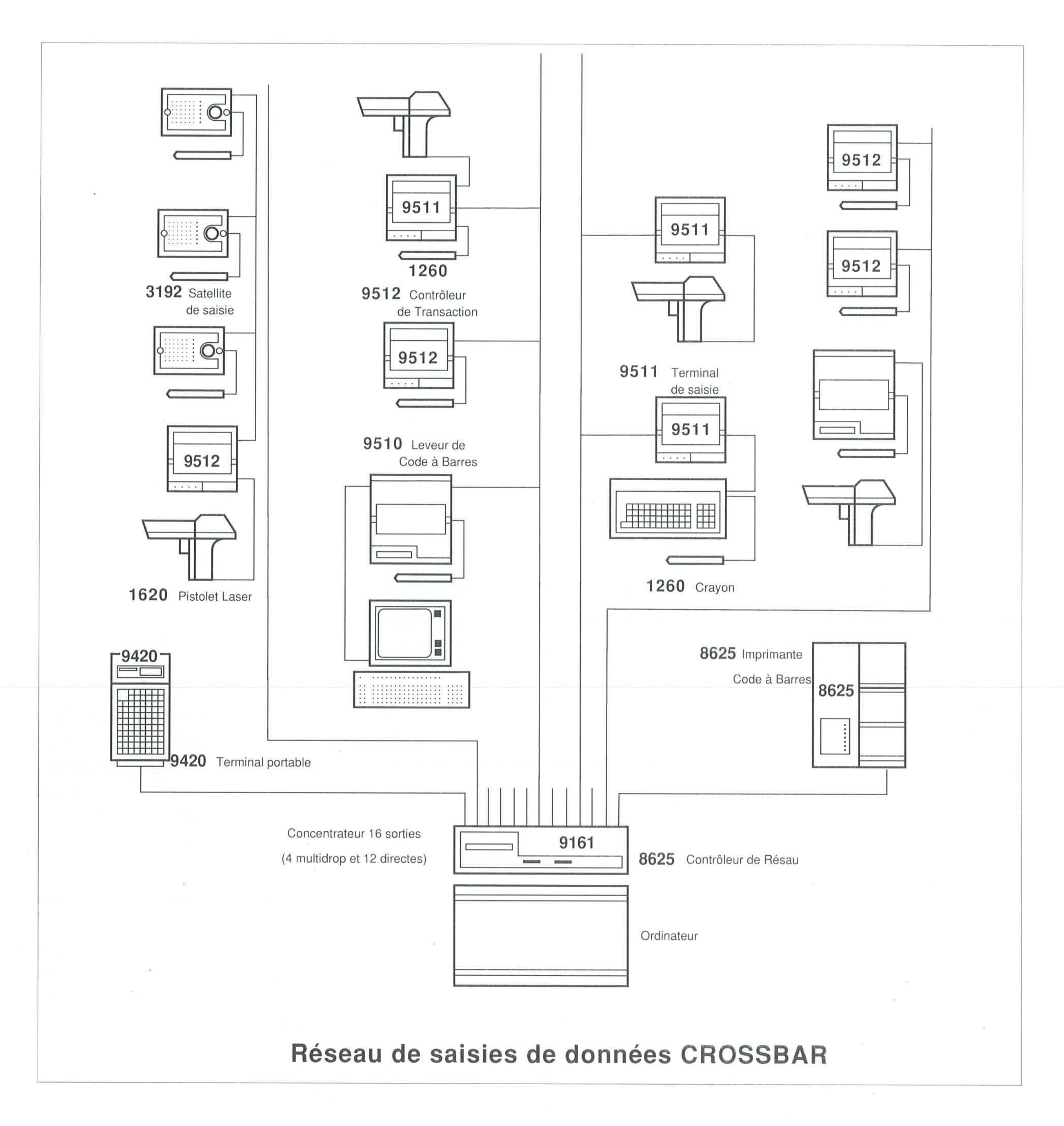
En allant plus loin dans l'utilisation de programmes locaux dans les terminaux, programmes locaux qui peuvent être téléchargés du site central pour être modifiés même en cours de journée sans interrompre les saisies, on peut concevoir un système où le flux d'informations à un instant donné devienne presque uniquement uni-directionnel, l'ensemble du réseau de saisie fonctionnant alors sans appel au système central.. Celui-ci, quand il le souhaitera, viendra interroger le réseau.

Cette façon de fonctionner est parfaitement illustrée par le réseau Crossbar mis au point par la société INTERMEC.

CROSS-BAR: LA SIMPLICITE

Crossbar est un cas tout à fait particulier. Destiné à être installé dans des ateliers ou des usines exploitant des terminaux codes barres pour faire la saisie des événements (début de fabrication, vérification d'assemblage, fin de fabrication, emballage, etc...), ce réseau industriel est d'une simplicité quasi-biblique, ce contrairement à des architectures transactionnelles plus lourdes (MAP, par exemple).

Le réseau se compose d'un concentrateur autonome intelligent, ayant son propre processeur (MC 68000) et



Réseau de saisies de données CROSSBAR

sa propre horloge interne et d'une ou plusieurs lignes deux paires torsadées blindées qui, dans le cas le plus simple, serpentent sur une distance maximum de 600 mètres par ligne dans l'atelier. Typiquement, on installe celle-ci dans le plafond. La liaison est de type RS 422 et permet un débit à relativement faible vitesse (9.600 ou 19.200 bps), compensé par la faible taille des paquets expédiés (15 caractères, en moyenne).

35 terminaux peuvent être connectés sur un bus. Le concentrateur va ensuite interroger ceux-ci adresse par adresse et vérifier dans la boîte aux lettres de chacun s'il a un ou plusieurs messages à transmettre. De son côté, si le concentrateur a, par exemple, un message pour A, il le déposera dans la boîte. En bref, ici c'est le concentrateur qui est maître du réseau, ce qui évite et l'emploi d'un protocole qui grèverait les temps de réponse en alourdissant les trames des paquets, et un système sophistiqué de gestion des collisions, les acquittements étant gérés par le concentrateur. On configurera toutefois le réseau en déclarant les adresses "vides" afin que le concentrateur ne perde pas son temps à les interroger. Ainsi dispose-t-on d'un système possédant sa propre mémoire (64 Ko dans le concentrateur dont 50 réservés aux tampons utilisateurs), d'une remarquable tolérance et dont les temps d'attente sont extrêmement courts, ceci sans qu'un logiciel spécifique ne soit nécessaire pour la capture des informations saisies par les terminaux codes barres.

Bibliographie:

La clé du code barres. Alain Macaigne. INTERMEC EUROPARC Créteil 1 allée des Cerisiers, 94042 Créteil cedex.